REMARKS

In the Office Action, the Examiner rejected claims 1-7 and 24 under 35 USC §103. The claims have been amended to further clarify the subject matter regarded as the invention. Claims 1-7 and 24-26 remain pending. The claim rejections are fully traversed below.

Reconsideration of the application is respectfully requested based on the following remarks.

REJECTION OF CLAIMS UNDER 35 USC §103

In the Office Action, the Examiner rejected the claims under 35 USC §103 as being unpatentable over Hayes et al, U.S. Patent No. 6,073,212, ('Hayes' hereinafter) in view of Hunt, U.S. Patent No. 6,192,398, ('Hunt' hereinafter), and further in view of Blumenau et al, U.S. Patent No. 6,260,120, ('Blumenau' hereinafter). This rejection is fully traversed below.

> To establish a prima face case of obviousness, three basic criteria must be met. First there must be some suggestion or motivation, either in the references themselves or in the knowledge generally available to one of ordinary skill in the art, to modify the reference or combine reference teachings. Second, there must be a reasonable expectation of success. Finally, the prior art reference (or references when combined) must teach or suggest all claim limitations. The teaching or suggestion to make the claimed combination and reasonable expectation of success must both be found in the prior art, and not based upon applicants disclosure. See MPEP § 2142.

Hayes discloses a system in which "read-only constraints preclude store operations directed to a cache line in third cache 32 from being serviced unless the requesting client such as **processor** 22-2 asks for permission first. For example under a MOESI cache coherency protocol, if a **processor** has a cached copy in an "exclusive" state, the processor can safely read and write to the data copy without checking other caches for another data copy or informing other **processors (in a multiprocessor system)** because the data copy is exclusive. When a **processor** has a "read-only" copy, it means that there is a potential that more than one data copy of the same memory location is kept in other **processor** caches in the system. See Hayes,

col. 6, lines 8-19; emphasis added. As can be seen, Hayes relates to the management of processor caches in a multi-processor system.

Applicants respectfully point out that Hayes neither teaches, suggests, or motivates operations upon HTTP requests, operations upon an HTTP cache such as an in-kernel HTTP cache, or operations regarding an HTTP daemon. See Hayes generally. Hayes neither discloses nor suggests managing a HTTP cache in a web server. Specifically, Hayes fails to disclose or suggest receiving a HTTP request, and therefore fails to disclose or suggest determining whether response data associated with the HTTP request is in a HTTP cache such as an in-kernel HTTP cache. While Hayes discloses checking various processor caches for a data copy and the need to inform other processors when data is written to an "exclusive" cached copy, Hayes neither discloses nor suggests requesting permission from a HTTP daemon in order to transmit data obtained from an in-kernel HTTP cache. Specifically, Hayes neither discloses nor suggests obtaining an advisory state indicating whether it is necessary to obtain permission from the HTTP daemon. In fact, the Examiner admits that Hayes fails to teach an advisory state indicating whether or not permission from a HTTP daemon needs to be obtained. Thus, with respect to claims 1 and 24, Hayes neither discloses nor suggests "when it is determined that response data associated with the HTTP request is in the in-kernel HTTP cache, obtaining an advisory state associated with the HTTP request from the in-kernel HTTP cache, the advisory state when in a first state indicating that it is necessary to obtain permission from a HTTP daemon to determine whether the response data can be transmitted and when in a second state indicating that the response data can be transmitted without obtaining permission from the HTTP daemon" or "transmitting the response data in accordance with the advisory state associated with the HTTP request."

The Examiner alleges that Hayes discloses obtaining an advisory state associated with the request from the HTTP cache. However, the Examiner admits that Hayes fails to teach obtaining an advisory state associated with the HTTP request from the HTTP cache, where the advisory state indicates whether it is necessary to obtain permission from the HTTP daemon to determine whether the response data can be transmitted. In no manner does Hayes disclose or suggest managing a HTTP cache such as an in-kernel HTTP cache in a web server. As described in the Summary of

Applicant's specification, it may be desirable to obtain permission from the HTTP daemon to transmit time sensitive information such as stock information that may potentially be outdated. While Hayes discloses cached data in an "exclusive" or "read-only" state, such information only serves to protect the accessibility of data in the "exclusive" state and prevents data from being overwritten in the "read-only" state. Hayes fails to disclose or suggest the problem associated with transmitting time sensitive information that may be outdated to a client, nor does Hayes suggest a solution to this problem. Moreover, Hayes neither discloses nor suggests a method of managing an in-kernel HTTP cache in a web server.

The Examiner admits that Hayes does not teach an advisory state indicating that response data is stored, or not, in the HTTP cache. Applicants reiterate that Hayes contains no teaching of any sort of operation, upon an HTTP cache such as an in-kernel HTTP cache. See Hayes generally.

Accordingly, Applicants respectfully point out that Hayes cannot teach, suggest or motivate what the Examner alleges in his discussion regarding Hayes. For example, Hayes cannot teach determining whether an HTTP cache copy or content (response data) associated with an HTTP request is in the in-kernel HTTP cache; obtaining an advisory state associated with the HTTP request from the in-kernel HTTP cache associtaed with the determined existing HTTP cached copy or HTTP cached content; or reading (retrieving) the response data associated with an HTTP request, in accordance with the advisory state associated with an HTTP request.

The Examiner appears to rely upon Hunt to show "a system/method related to the management of a cache (124) in a cache memory system (col 3/lines 45-49) in a web server (116-118) (col 3/lines 30-44, 52-58, col 4/lines 46-48, Fig. 1)" to make up for the deficiencies of Hayes. See Office Action Page 1. Specifically, the Examiner appears to rely upon Hunt to show "an indication ("state") indicating that the response data is stored or not in the ("HTTP") cache, including whether response data associated with the HTTP request is in the cache (step 508, Fig. 5A, col 6/lines24-28). See Office Action Page 2. The Examiner goes on to cite step 518 of Fig. 5A, col 6/lines 29-32, 54-61, step 510 of Fig. 5A to show "obtaining an advisory state associated with the HTTP request from the HTTP cache associated with the cached response." See Office Action Page 2. The Examiner also cites Fig. 5B, step 514 to

show "transmitting (loading) response data in accordance withg the advisory state associated with the HTTP request." See Office Action Page 2.

Applicants respectfully disagree with the Examiner's allegations. At the outset, Applicants kindly point to the title of Hunt, "REMOTE/SHARED BROWSER CACHE". As to the citations of Figs. 5A and 5B, Applicants respectfully point out that the processes laid out in those figures begins item 502, "Browser Started", indicate that the process described relates to a browser cache. Applicants respectfully contend that Hunt relates to activities upon a web browser cache. See Hunt generally. While Hunt does use the term 'server', Applicants note that this is a generic discussion of storage devices (see Hunt columns 2 and 3); Hunt does not discuss managing a cache upon a server, much less a web server HTTP cache such as an in-kernel HTTP cache, as the Examiner appears to allege. See Hunt generally.

Hunt fails to cure the deficiencies of the primary reference. Hunt teaches a system in which storage devices (e.g., cache) provide remote storage for user units. See Hunt, col. 3, lines 45-49. Specifically, multiple remote/local browser caches are used in combination with one another. See col. 5, lines 37-40. When a request is received to load a specific Web page, the process determines whether an acceptably recent copy of the requested page is cached in a browser cache. If a sufficiently recent version is not cached in the browser cache, the in-progress flag associated with the requested page in the cache contents information for each browser cache which should contain the requested page is set. The requested page is then loaded or updated, and cached, which may include a lookup to a policy for caching at various local and/or remote shared browser caches. See col. 6, lines 20-42. If a requested page is cached in a browser cache accessible to the browser, the process determines whether an update of the requested page in the browser cache is required. See col. 6, lines 54-58. Thus, Hunt teaches managing memory in a shared browser cache. Hunt fails to disclose or suggest managing a HTTP cache (e.g., in-kernel HTTP cache) in a web server. Moreover, Hunt fails to disclose or suggest obtaining an advisory state indicating whether it is necessary to obtain permission from the HTTP daemon in order to transmit response data obtained from an in-kernel HTTP cache. It follows that Hunt fails to disclose or suggest transmitting response data in a web server in accordance with the advisory state associated with a HTTP request.

Therefore, because Hunt does not teach, suggest or motivate operations on, or activities relating to, HTTP requests to an in-kernel HTTP cache of a web server, Hunt cannot teach any elements of Claim 1.

Neither Hayes nor Hunt discloses or suggests an "advisory state when in a first state indicating that it is necessary to obtain permission from a HTTP daemon to determine whether the response data can be transmitted and when in a second state indicating that the response data can be transmitted without obtaining permission from the HTTP daemon" as recited in claims 1 and 24. The Examiner seeks to cure the deficiencies of Hayes and Hunt with Blumenau. Specifically, the Examiner appears to rely upon Blumenau to allege a teaching of "a system/method relating to data processing networks and data storage subsystems, including a flag ("advisory state") which when set (first state) the private/shared flag indicates that all resources are private and no permission is needed from a lock manager before the host controller port can accessing any of the resources its assigned resource list". See Office Action Page 2.

Blumenau discloses a storage controller for controlling access to data storage. See Abstract. Specifically, a large number of hosts can access volumes of data storage. See col. 1, lines 9-14. Blumenau does disclose a flag. However, the flag disclosed in Blumenau does not operate in the same manner as the advisory state of claims 1 and 24. Rather, Blumenau discloses a private/shared flag that, when set, indicates that all of the logical volumes in the list are private and no permission is needed from a lock manager before the host controller port can access any logical volume in its assigned volume list. See col. 15, lines 11-15.

Applicants respectfully contend that Blumenau adds nothing to either Hunt, Hayes, or their combination, to substantiate the Examiner's allegation. Blumeneau does not teach, suggest, or motivate, operations upon HTTP requests, operations upon an HTTP cache, operations regarding an HTTP daemon, or operations upon HTTP requests. Hayes and Hunt suffer the same deficiency, as pointed out above. Therefore, Blumenar cannot teach suggest or motivate any sort of an advisory state acting upon the same.

Blumenau neither discloses nor suggests managing a HTTP cache in a web server. Thus, Blumenau fails to disclose or suggest managing an in-kernel HTTP

cache in a web server. In fact, Blumenau appears to operate similarly to Hayes. While Hayes relates to "read-only" and "exclusive" states in a multi-processor system, Blumenau relates to the "locking" of data in a multi-processor system. While Blumenau does disclose requesting permission from a lock manager to access a logical volume, Blumenau fails to cure the deficiencies of Hayes and Hunt. Specifically, none of the cited references, separately or in combination, discloses or suggests requesting permission from a HTTP daemon in order to transmit data. In fact, Hayes and Blumenau teach that if data is accessible, it may be transmitted once it has been accessed. As such, these references teach away from requesting permission from a HTTP daemon in order to transmit HTTP response data that has already been obtained from a HTTP cache or can be obtained from a HTTP cache.

In addition, Blumenau and Hayes appear to solve the same problem (i.e., to control access to stored data in a multi-processor system). This is accomplished in Hayes through the use of "read-only" and "exclusive" states, and in Blumenau through a private/shared flag. Stated another way, Hayes solves this problem through the use of different states, while Blumenau solves this problem through the use of a flag. Since both of these references appear to independently solve the same problem, there fails to be a motivation to combine these references.

It is also important to note that none of the references disclose or suggest managing a HTTP cache in a web server, or a HTTP daemon that controls the transmission of data and/or caching of data in the manner claimed. Thus, the references fail to suggest the desirability and obviousness of combining the cited references for the desired purpose (i.e., managing a HTTP cache such as an in-kernel HTTP cache in a web server). Specifically, since each of the references fails to disclose a process of managing a HTTP cache in a web server, the references fail to suggest the desirability and obviousness of combining the cited references to manage a HTTP cache such as an in-kernel HTTP cache in a web server. In addition, since none of the references discloses or suggests a process of managing a HTTP cache in a web server or pertinent to the problem of managing a HTTP cache in a web server, these references are arguably non-analogous art, since they are not reasonably pertinent to the problem that the claimed invention solves (e.g., ensuring that accurate data is transmitted to a client by a web server in response to a HTTP request).

Even if the references were combined, the combination would fail to achieve the desired result. Hayes relates to a multi-processor system including processor caches, while Hunt relates to a shared web browser cache. It would generally be undesirable to manage a web browser in the same manner as a processor cache in a multi-processor system. In addition, none of the references discloses or suggests managing an in-kernel HTTP cache in a web server, or more specifically, requesting permission from a HTTP daemon to transmit data from a HTTP cache associated with a HTTP request. Thus, the combination would be unsatisfactory for its intended purpose (i.e., managing a HTTP cache in a web server). As such, there would fail to be a reasonable expectation of success.

The invention of claims 1 and 24 transmits the data in accordance with an advisory state, as claimed. Specifically, when data is obtained from a HTTP cache, an advisory state associated with the HTTP request is also obtained. As described above, the advisory state when in a first state indicates that it is necessary to obtain permission from a HTTP daemon to determine whether the response data can be transmitted and when in a second state indicates that the response data can be transmitted without obtaining permission from the HTTP daemon. The cited references, separately or in combination, fail to disclose or suggest obtaining an advisory state indicating whether it is necessary to obtain permission from the HTTP daemon in order to transmit response data in an in-kernel HTTP cache in a web server.

The dependent claims depend from one of independent claims 1 and 24 and are therefore allowable for at least these reasons. In addition, the dependent claims recite additional limitations that further distinguish them from the cited references. For instance, claim 3 recites "transmitting the response data without modifying the response data in the HTTP cache when the advise state is in a first state." In other words, the response data is temporary, and therefore may be transmitted to the client but not stored in the in-kernel HTTP cache.

As shown above, neither Hayes, Hunt nor Blumeneau, nor their combination suggest or motivate to modify any of the cited references, or to combine their teachings to arrive at the Examiner's proposed combination. Because none of the references teaches operation upon an in-kernel HTTP cache in a web server, or

operation upon HTTP requests related thereto, there can be no expectation of success to arrive at the Examiners alleged conclusion. Finally, also for reasons set forth above, none of the references relied upon by the Examiner, either alone or in combination, teaches all of the claim limitations as alleged by the Examiner. Accordingly, Applicants contend that the rejection is unsupported by the art and should be withdrawn.

The cited references, separately or in combination, fail to disclose the claimed invention. Hence, it is submitted that the independent claims are patentable over the cited art. The additional limitations recited in the independent claims or the dependent claims are not further discussed as the above discussed limitations are clearly sufficient to distinguish the claimed invention from the cited art. Thus, it is respectfully requested that the Examiner withdraw the rejection of the claims under 35 USC §103.

If there are any issues remaining which the Examiner believes could be resolved through either a Supplemental Response or an Examiner's Amendment, the Examiner is respectfully requested to contact the undersigned attorney at the telephone number listed below.

Applicants hereby petition for an extension of time which may be required to maintain the pendency of this case, and any required fee for such extension or any further fee required in connection with the filing of this Amendment is to be charged to Deposit Account No. 50-0388 (Order No. SUN1P701).

Respectfully submitted,
BEYER, WEAVER & THOMAS, LLP

Elise R. Heilbrunn
Reg. No. 42,649

BEYER, WEAVER & THOMAS, LLP
P.O. Box 778
Berkeley, California 94704-0778
Tel. (510) 843-6200